Implementation of Hybrid Control for Motor Drives

Xuefang Lin-Shi, Florent Morel, *Student Member, IEEE*, Ana M. Llor,
Bruno Allard, *Senior Member, IEEE*, and Jean-Marie Rétif

***Abstract*—This paper presents the implementation of a hybrid-control strategy applied to a permanent-magnet synchronous-motor (PMSM) drive. Hybrid control is a general approach for control of a switching-based hybrid system (HS). This class of HS includes a continuous process controlled by a discrete controller with a finite number of states. In the case of ac motor drives, in contrast to conventional vector control like proportional–integral control or predictive control, where the inverter is not taken into account by the controller, hybrid control integrates the inverter model and considers the state of the inverter as a control variable. It allows to obtain faster torque dynamics than vector-control algorithms. The hybrid control algorithm requires both computing velocity for real-time implementation and code flexibility for management of low-performance functions and analog–digital interfaces. Codesign appears as a promising methodology for partitioning hybrid-control algorithm between software (flexible) and hardware (velocity) while taking care of overall time constrains. In this paper, the implementation of hybrid-control algorithm for a PMSM drive is performed through a codesign approach on an Excalibur board, embedding a CPU-core (Nios-2 by Altera) inside an APEX20KE200EFC484-2X field-programmable gate array. The partitioning of software and hardware parts is explained. Experimental results show the effectiveness of the implementation. Performances, advantages, and limitations are discussed.**

***Index Terms*—AC motor drives, control, dynamic hybrid system (HS), field-programmable gate array (FPGA), hardware–software codesign.**

I. Introduction

THERE ARE a large number of industrial applications where the system consists of a continuous plant with a discrete-event controller. In ac-motor drives, the continuous part is the synchronous or asynchronous machine, and the voltage inverter corresponds to the discrete part.

In ac-drive systems, fast current and torque responses are necessary. The development of high-performance control for the ac-machine drive has motivated considerable attention in recent years. In the mid-1980s, the direct torque control technique (DTC) was developed [1], [2]. The basic principle of DTC is to select the appropriate stator voltage vectors from a table, according to the signs of the errors between the references of torque and stator flux and their estimated values, respectively [3], [4]. DTC advantages are low machine parameter dependence and a fast dynamic torque response. This technique involves a switching table and hysteresis controllers. Stator-flux vector and torque estimations are necessary [5].

In this paper, a hybrid-control technique is proposed. It is a general approach for control of a class of hybrid systems (HS): controlled-switching HS [6]. This class of HS generally consists of a continuous-time plant with a finite discrete-event controller. The HS under consideration is a permanent-magnet synchronous motor (PMSM) combined with a two-level three-phase voltage inverter. Based on the use of a simple hybrid model including the motor and the inverter, hybrid-control algorithm calculates the direction evolution of the current vector for all possible switching states of the inverter. The switching state, which minimizes a given cost function, is selected. The selected inverter state is applied during a calculated time duration.

Both DTC and hybrid control determine directly an inverter switching state, and similar torque dynamics can be obtained. However, significant differences between hybrid control and DTC can be noted: DTC takes heuristic decisions according to a selection table; hybrid control is based on a formal representation of the whole inverter-machine system. DTC uses hysteresis controllers to minimize torque and flux errors; hybrid control minimizes errors between measured currents and reference currents, so no hysteresis controller neither observer are necessary. Moreover, hybrid control is a general approach for any controlled-switching HS where the discrete control states can take only a finite number of values.

Hybrid-control algorithm requires computation velocity like DTC algorithm to reduce current ripples. Literature covers implementation in all field-programmable gate arrays (FPGAs) of motor control schemes [7]–[10]. High performance is demonstrated, but lack of flexibility is also pointed out as a severe limitation from the industrial point-of-view. Reusability of parts of the implementation is questioned, as it is not easy to retrieve the algorithm sequences. In addition, digitally controlled systems with digital signal processors (DSPs) have been presented in many publications on drive-control technology [11], [12]. In [13], authors express the limitations of DSP in terms of velocity for modern control algorithms if cost is mandatory. Codesign has been introduced as a set of methodologies for partitioning advanced control algorithm between software and hardware parts [14], [15]. It offers the engineer to apprehend a wide space of solutions, where a full DSP solution or a full FPGA solution is two extremities, and the ability to select a suitable tradeoff with regard to performances and cost. Examples of codesigned control systems have been presented in the study in [15] and [16].

The implementation of the proposed hybrid control algorithm is performed through a codesign approach. The algorithm

Manuscript received April 24, 2006; revised December 6, 2006.

The authors are with Laboratoire Ampère (UMR CNRS 5005), Institut National des Sciences Appliquées de Lyon, F-69621 Villeurbanne Cedex, France (e-mail: xuefang.shi@insa-lyon.fr).

Digital Object Identifier 10.1109/TIE.2007.898303

has been implemented in an Excalibur board embedding a CPU-core (Nios-2 by Altera) inside an APEX20KE200EFC484-2X FPGA. It is an industrial prototyping board. Afterward, a just-what-is-required FPGA could be selected from an industrial production point of view. The FPGA circuit offers a practical silicon area lower than an industrial DSP but does not embed A/D converters either. Footprint on a printing circuit board is not necessarily in favor of the FPGA solution, but the DSP solution does not let the engineer tailor suitable A/D converters. External A/D converters require suitable interface that DSP already includes. This issue is not discussed here, but it may be safely advanced that FPGA including A/Ds will be introduced shortly [17].

Codesign takes advantage of parallelism offered by the algorithm. The efficient hardware and software partitioning allows to run the control algorithm in less than 10 μs of loop-cycle. Experimental results show performances of the proposed control and implementation solution.

This paper is organized as follows. Section II explains the principle of the hybrid-control algorithm for a PMSM. Section III covers the codesign implementation of the controller for a 1.6-kW PMSM drive. Experimental results are presented in Section IV. The control performances are evaluated by comparative studies with respect to a vector control and a DTC control, both implemented in an oversized DSP [18].

II. Hybrid Control Principle

A. Hybrid Model

As aforementioned, the proposed hybrid control addresses a class of HS composed of a continuous process controlled by an energy modulator which has a finite number of switching states or topologies. In the case of a PMSM drive system, the PMSM constitutes the continuous process and the inverter corresponds to the energy modulator. A hybrid model including both the PMSM and the inverter can be established.

A PMSM consists of three-phase stator windings and permanent magnets either mounted on the rotor surface (surface-mounted PMSM) or buried inside the rotor (interior PMSM). The continuous state equations of a surface-mounted PMSM, which is written in the stator flux reference frame, can be expressed as

$$\begin{bmatrix} \frac{dI_{\mathrm{sd}}(t)}{dt} \\ \frac{dI_{\mathrm{sq}}(t)}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_s} & \omega_r(t) \\ -\omega_r(t) & -\frac{R_s}{L_s} \end{bmatrix} \cdot \begin{bmatrix} I_{\mathrm{sd}}(t) \\ I_{\mathrm{sq}}(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L_s} & 0 & 0 \\ 0 & \frac{1}{L_s} & -\frac{\omega_r(t)}{L_s} \end{bmatrix} \cdot \begin{bmatrix} V_d(t) \\ V_q(t) \\ \Phi \end{bmatrix} \quad (1)$$

where I_{sd} and I_{sq} are the d- and q-axes stator currents, R_s is the stator phase resistance, L_s is the stator phase inductance, ω_r is the rotor angular velocity, V_d and V_q are the d- and q-axes voltages, and Φ is the flux established by permanent magnets.

The stator voltages V_d and V_q depend on the switching states of the three-phase inverter.

The switching states of the inverter, when it is considered as ideal, can be described by discrete variables u_A, u_B, and u_C in the following form (Fig. 1):

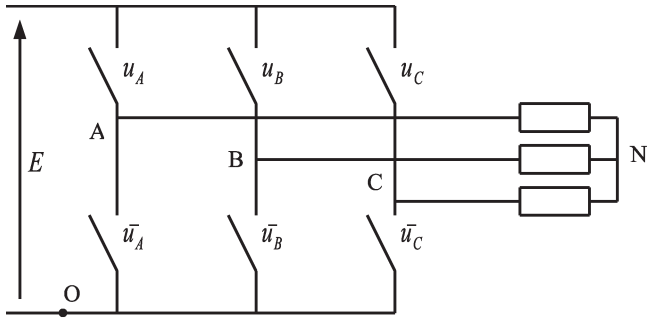

Fig. 1. Simplified schematic of the PMSM drive.

TABLE I
Inverter Switching States

j	0	1	2	3	4	5	6	7
u_A	0	1	1	0	0	0	1	1
u_B	0	0	1	1	1	0	0	1
u_C	0	0	0	0	1	1	1	1

$$u_A = \begin{cases} 0 \Leftrightarrow V_{A0} = 0 \\ 1 \Leftrightarrow V_{A0} = E \end{cases}$$
$$u_B = \begin{cases} 0 \Leftrightarrow V_{B0} = 0 \\ 1 \Leftrightarrow V_{B0} = E \end{cases}$$
$$u_C = \begin{cases} 0 \Leftrightarrow V_{C0} = 0 \\ 1 \Leftrightarrow V_{C0} = E \end{cases}. \quad (2)$$

The winding voltages can then be written as

$$\begin{bmatrix} V_{\mathrm{AN}}(t) \\ V_{\mathrm{BN}}(t) \\ V_{\mathrm{CN}}(t) \end{bmatrix} = \frac{E}{3} \cdot \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} u_A(t) \\ u_B(t) \\ u_C(t) \end{bmatrix}. \quad (3)$$

In the same way, these voltages can be expressed in the $\alpha\backslash\beta$ stator reference frame and multiplied by a rotation matrix to obtain the voltage expressions in the Park reference frame

$$\begin{bmatrix} V_d(t) \\ V_q(t) \end{bmatrix} = E\sqrt{\frac{2}{3}} R(\theta_r) \cdot \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} u_A(t) \\ u_B(t) \\ u_C(t) \end{bmatrix} \quad (4)$$

where θ_r is the angular rotor position, and the matrix R is defined as

$$R(\theta_r) = \begin{bmatrix} \cos\theta_r & \sin\theta_r \\ -\sin\theta_r & \cos\theta_r \end{bmatrix}. \quad (5)$$

The correspondence between a switching state number (j) and the values of u_A, u_B, and u_C is given in Table I.

Among the eight possible states of the inverter, two of them (corresponding to $j = 0$ and $j = 7$) lead to the same output voltages ($V_d = V_q = 0$). Therefore, for the sake of simplicity, the case $j = 0$ will not be considered.

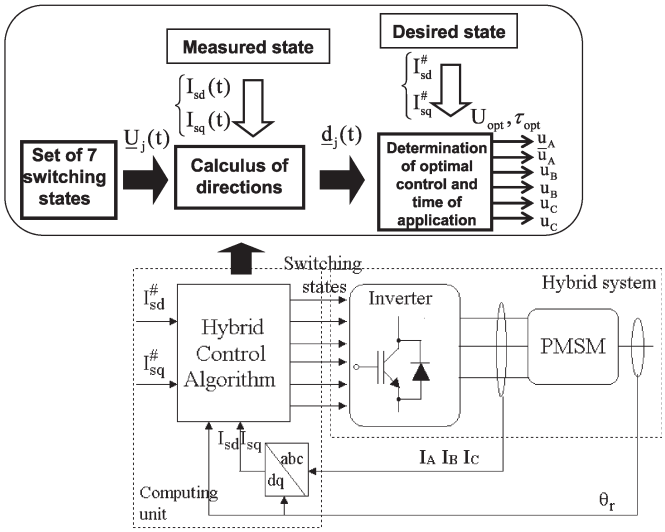

Fig. 2. Hybrid control algorithm for PMSM drive.

Finally, substituting (4) into (1), the model of the PMSM and its inverter can be written as

$$\dot{\underline{X}}(t) = A\left(\omega_r(t)\right) \cdot \underline{X}(t) + B_1\left(\omega_r(t), \theta_r(t)\right) \cdot \underline{U}(t) + B_2(\omega_r(t)) \cdot \Phi \quad (6)$$

where $\underline{X}(t) = [I_{\rm sd}, I_{\rm sq}]^t$ is the continuous state vector, and $\underline{U}(t) = [u_A, u_B, u_C]^t$ is the control vector which can take one of the seven switching states in Table I, except state zero as explained.

Assuming that ω_r and θ_r are unchanged during a very short time τ, (6) can be written as

$$\begin{aligned}\underline{X}(t+\tau) &= [I + \tau \cdot A\left(\omega_r(t)\right)] \cdot \underline{X}(t) \\ &\quad + \tau \cdot B_1\left(\omega_r(t), \theta_r(t)\right) \cdot \underline{U}(t) \\ &\quad + \tau \cdot B_2\left(\omega_r(t)\right) \cdot \Phi \\ &= F\left(\omega_r(t)\right) \cdot \underline{X}(t) + H\left(\omega_r(t), \theta_r(t)\right) \cdot \underline{U}(t) \\ &\quad + E\left(\omega_r(t)\right) \cdot \Phi. \end{aligned} \quad (7)$$

Therefore, for each control vector $\underline{U}_j(t)$ $(j = 1, \ldots, 7)$, the corresponding state vector $\underline{X}_j(t+\tau)$ can be calculated if $\underline{X}(t)$, $\omega_r(t)$, and $\theta_r(t)$ are known quantities.

B. Hybrid-Control Algorithm

The principle of hybrid control for PMSM drive is summarized in Fig. 2. It is based on the hybrid model (7). It determines the optimal inverter switching state which minimizes errors between measured and reference state variables.

For a desired reference state vector $\underline{X}^{\#} = [I^{\#}_{\rm sd}, I^{\#}_{\rm sq}]^t$, hybrid control proceeds as follows inside each computation iteration (loop-cycle).

1) Measure the stator currents and angular rotor position, then calculate the rotor angular velocity ω_r, the currents $I_{\rm sd}(t)$, $I_{\rm sq}(t)$, the state matrices $F(\omega_r)$, $H(\omega_r, \theta_r)$, and finally, $E(\omega_r)$.
2) Calculate the direction of the state vector evolution defined by $\underline{d}_j = \underline{X}_j(t+\tau) - \underline{X}(t)$ for the seven possible control vectors, $\underline{U}_j(t)(j = 1, \ldots, 7)$.

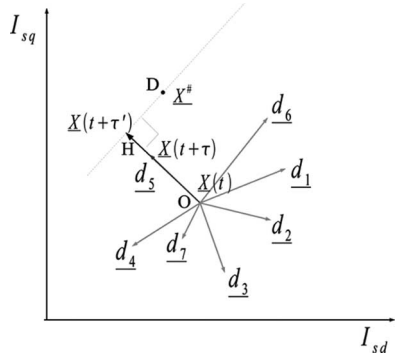

Fig. 3. Typical state vector evolution in state space.

3) Evaluate the cost function. Determine the optimal control vector $\underline{U}_{\rm opt}$ that corresponds to the minimal cost function. There are many ways to define a cost function. The angle between directions $\underline{d}_j$ and $|\underline{X}^{\#} - \underline{X}(t)|$ is selected here as the cost function.
4) Calculate the time duration $\tau_{\rm opt}$ for $\underline{U}_{\rm opt}$. On the one hand, this calculus is based on the assumption that, during a short time of application of $\underline{U}_{\rm opt}$, the evolution of the state vector is a straight line, and its amplitude is proportional to the time of application. This assumption limits $\tau_{\rm opt}$ to a maximum value $\tau_{\max}$. On the other hand, due to technology limitations of converters (inverter dead-time, velocity of A/D converters, ...) and of the CPU performance (instruction clock), $\tau_{\rm opt}$ must be larger than the duration of the algorithm loop cycle and the inverter dead-time, so a minimum time of application $\tau_{\min}$ must be respected. Finally, the application time $\tau_{\rm opt}$ can be expressed as

$$\begin{aligned}&\left|\underline{X}(t+\tau') - \underline{X}^{\#}\right| = \min_{\Delta t} \left|(t+\Delta t) - \underline{X}^{\#}\right| \\ &\quad \text{if } \tau' < \tau_{\min}, \text{ then } \tau_{\rm opt} = \tau_{\min} \\ &\quad \text{if } \tau' > \tau_{\max}, \text{ then } \tau_{\rm opt} = \tau_{\max} \\ &\quad \text{else } \tau_{\rm opt} = \tau'.\end{aligned}$$

Fig. 3 illustrates an example of the state vector evolution in the state space $[I_{\rm sd}, I_{\rm sq}]$. Points O and D are, respectively, the measured and desired states. Seven directions of the state vector evolutions are represented by vectors $\underline{d}_j (j = 1, \ldots, 7)$ issued from O. In this example, the direction $\underline{d}_5$ has the minimum angle with $|\underline{X}^{\#} - \underline{X}(t)|$, so the optimal control vector will be $\underline{U}_5$, corresponding to the switching state $[u_A, u_B, u_C]^t = [0, 0, 1]^t$. The time of application of $\underline{U}_5$, τ', is calculated easily from the distance between points O and H.

A key difference between hybrid control and a classical vector control is that the control vector $\underline{U}_{\rm opt}$ of hybrid control corresponds to one of the seven states of the inverter. It is not a switching sequence as for the pulsewidth modulation (PWM). For each computation iteration, only one state is applied, as for a DTC control. If the reference values are changed between two computing cycles, a new appropriate control state and the corresponding application time will be determined for the next computation cycle. This is a common feature to any digital control scheme. As the application time is bounded by $\tau_{\max}$, the external influences can be taken into account quickly.

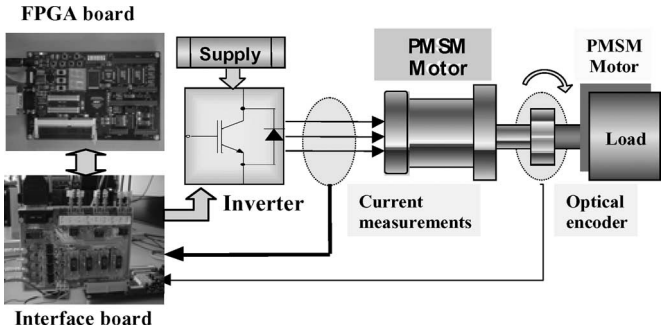

Fig. 4. Experimental test bench.

TABLE II
RATING VALUES OF TESTED PMSM

1.56 kW	5.5 Nm	3000 rpm	3 pairs of poles
R_s=2.06 Ω	L_s=9.15 mH	Φ=0.29 Wb	

As there are only seven possible directions for the state vector, the reference point could not be exactly reached. When the state $\underline{X}(t)$ is close to the reference point $\underline{X}^{\#}$, as $\tau_{\rm opt}$ is limited to $\tau_{\rm min}$, the state vector $\underline{X}(t+\tau_{\rm min})$ could then oscillate around the reference point. The smallest the $\tau_{\rm min}$ value, then the lowest the current ripple. Therefore, a high-performance computing unit is necessary because $\tau_{\rm min}$ value is limited by data conversions and computation time which depend directly on the computing unit performance.

III. EXPERIMENTAL DRIVE CONFIGURATION AND REAL-TIME IMPLEMENTATION

A. Test Bench

The main components of the laboratory-scale experimental test bench are pictured in Fig. 4.

A 1.6-kW PMSM (SMV UM from Leroy–Sommer) is used with a 4096-pulse incremental encoder. Another identical PMSM is used as a torque generator, with a nominal torque of 5.5 Nm. The characteristics of the motor are given in Table II.

A commercial 15-kW three-phase inverter based on insulated-gate bipolar transistors is supplied by a voltage source XANTREX which provides 300 V with current limitation of about 10 A. The dead time is fixed at 3 μs due to the oversized inverter devices. Three LEM current sensors (LEM LA 100P) are used.

B. Control Unit

As explained previously, hybrid-control algorithm requires velocity for real-time implementation; moreover, the management of data conversion interfaces (current and angular position acquisitions and control output) demands flexibility. A codesign approach has been implemented using the Altera–Excalibur development kit including the Nios 2.0 version. The objective is to analyze the complete control algorithm and to define which parts suits hardware implementation by taking advantage of parallelism. Altera Quartus suite [19] has been used along with Mentor Graphics.

TABLE III
FPGA DEVICE FEATURES

FPGA device features	
Maximum system gates	526.000
Logic elements	8.320
Embedded system blocks	52
Maximum RAM bits	106.496
Maximum macro cells	832
Maximum user I/O pins	376
Phase-locked loops (PLL's)	2

Altera software offers the design of a user-defined processor FPGA-advantage tools (Nios core) [20]. This embedded processor is optimized for Altera programmable logic and system-on-a-programmable-chip (SOPC). It can be combined with user logic and programmed into an FPGA using SOPC builder software [21].

The Nios development board provides a hardware platform to develop embedded systems based on Altera APEX devices. The board includes an FPGA chip Apex20K200E, which offers practical silicon area lower than a standard DSP, but does not embed A/D converters. The device features used in the test bench are listed in Table III.

A suitable analog-and-digital hardware interface board has been added to the Excalibur board. It includes A/D converters (AD7892-1) for current acquisition, data-link buffer (4050N) for the encoder, and optocoupler circuits (T1521, R2521) for electrical isolation between the power drive and the FPGA board. The AD7892 converter is a 12-b A/D converter with a conversion time lower than 1.5 μs. This data-rate conversion is modest and has been chosen to demonstrate the efficiency of codesign, which takes care of such limitations through the algorithm partitioning. If A/D converters inside a DSP circuit feature such low performances, it would be practically impossible to implement any fast-loop algorithm.

C. Algorithm Implementation

The hybrid algorithm is sliced in elementary functions (Fig. 5). Each function is described from hardware viewpoint using Very-large scale integration Hardware Description Language, VHDL (IEEE 1076). Functions are naturally implemented separately hence in parallel. The FPGA resources are large enough to implement the individual blocks in hardware plus the embedded processor. Since velocity is a priority, the performances of the Nios core are not completely exploited, and hardware implementation has been preferred for several functions that could have been dedicated to the processor. The management of a serial link with the host computer and the management of the interface board switches are performed by software inside the FPGA embedded processor as well as the A/D converter logic control. The Nios core is a 16-b processor with 16-b input–output ports for real-time visualization of program variables via the serial link. The silicon area is then saved for other hardware blocks and a finite-state machine can overtake the sequencing of all the other operations.

The execution time and FPGA area for major functions are given in Table IV. In order to reduce $\tau_{\rm min}$, one part of current-independent functions is computed during the current

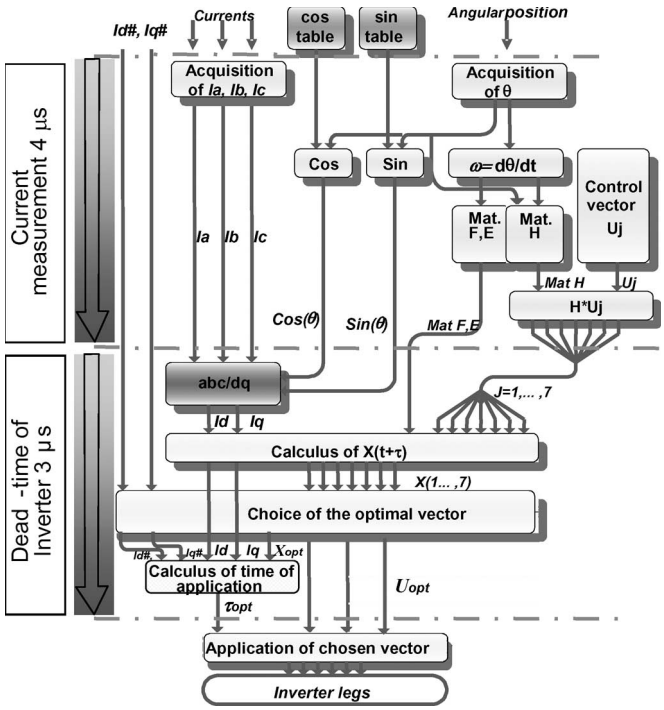

Fig. 5. Elementary functions for the hybrid control algorithm.

TABLE IV
EXECUTION TIME OF SOME BLOCK FUNCTIONS

	Exec.time(µs)	Logic elements
Forward Clarke Transformation	0.028	277 (3.3%)
Forward Park Transformation	0.035	1219 (14.7%)
Calculation electric angle	0.055	171 (1.4%)
Calculation of angular speed	0.03	136(1%)
Calculation of H*U_j	0.03	800 (9%)
Acquisition of currents	2.9	91 (1.1%)

acquisition (4 μs); another part of the functions is computed during the inverter dead-time (3 μs). Finally, the minimum time of application has been set to 10 μs.

VHDL codes are compiled, and a digital synthesis is performed using MentorGraphics ModelSim [22] and Leonardo Spectrum Tools [23]. Full C codes of software part and VHDL codes are available in [24].

IV. EXPERIMENTAL RESULTS

For an experimental comparative study, a classical vector-control technique and a DTC control are implemented. All experiments have been performed with the same experimental components as the hybrid control except for the DSP board. Data are recorded with the same measurement equipment and the same sampling period (200 μs).

For hybrid control, $\tau_{\min}$ has been set to 10 μs, and $\tau_{\max}$ has been limited to 100 μs. The vector control, using two proportional-integral (PI) discrete-time current controllers for $I_{\rm sd}$ and $I_{\rm sq}$, is implemented in a DSpace DS1104 board with the Simulink environment. Each controller is determined by pole assignment when considering a discrete-time first-order dynamics. As the process has a dynamics of 4.5 ms, the closed-loop dynamics corresponds to a time constant of 1 ms. The

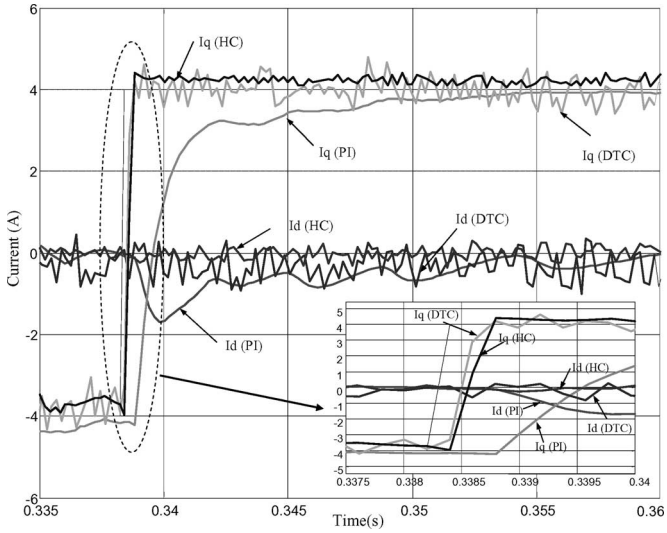

Fig. 6. Current waveforms during transient from a negative to a positive torque value.

controller sampling period has been set to 0.2 ms. The PWM switching frequency of the PI-controlled inverter has been set to 10 kHz. DTC control is implemented in C language on the same DSP as the vector control. The computation period has been fixed to 28 μs (limited by computation duration). Neither DTC nor hybrid control need PWM vector modulation.

The minimization of Joule power losses leads to maintain the current $I_{\rm sd}$ equal to zero while the electromagnetic torque is proportional to current $I_{\rm sq}$. Therefore, for hybrid control and vector control, the reference currents are chosen as $I_{\rm sd}^{\#} = 0$ A and $I_{\rm sq}^{\#}$ proportional to the desired torque. For DTC control, the references are directly the stator flux linkage reference and torque reference.

Experimental results in Fig. 6 correspond to a transient response from a negative current reference ($I_{\rm sq}^{\#} = -4$ A) to a positive reference ($I_{\rm sq}^{\#} = +4$ A) for hybrid and vector control. This experiment corresponds to a torque reference step from -3.48 to 3.48 N $\cdot$ m for DTC control. The speed changes from -1060 to 1060 r/min. No speed loop is used.

It can be seen that the q-axis current, which represents the electromagnetic torque, has a time response lower than 400 μs for DTC and hybrid control (Fig. 6, zoom). For vector control, the step change of q-axis current introduces a variation of d-axis current. This cross-coupling effect between $I_{\rm sd}$ and $I_{\rm sq}$ increases the time response of the q-axis current, which is then larger than 10 ms. The performance of DTC and hybrid control are similar but superior to PI controller, since they result in a shorter rise time and produces neither overshoot nor cross-coupling effects.

However, d- and q-axes current oscillations during steady-state operation are larger with both DTC and hybrid controls than with vector control. This is due to hysteresis controllers and limitation of the minimum sampling period for DTC and to the minimum application time for hybrid control as aforementioned (Section II-B).

Small errors in current are to be found with hybrid control in steady-state operation. For accurate applications, hybrid control

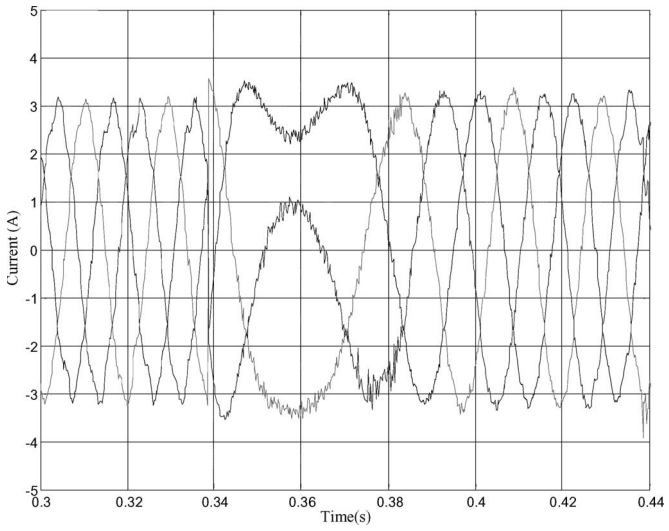

Fig. 7. Stator currents evolution during transition from negative rated torque to the positive one for hybrid control.

will be used inside a speed-control loop, where the static error can be compensated by the robust controller of the speed loop.

Fig. 7 pictures the transient evolution of the three stator currents with hybrid control. It can be noted that there is no over-current during torque and rotation-direction inversions. This is an interesting feature for practical implementation: a current regulator is not necessary for the hybrid control since the algorithm minimizes naturally the error between measured and reference currents.

Similar results have been obtained in many other different transient conditions.

V. Conclusion

The hybrid control presented here is a general approach for any system constituted of a continuous process and a finite number of discrete control states. In the context of machine control by means of an inverter, the hybrid control realizes the torque regulation dealing directly with the inverter switches. Very good results with hybrid control are obtained for PMSM drive and also for winding rotor synchronous machines and asynchronous machines not presented here. The superior transient performances of the proposed control algorithm over other conventional algorithms, like vector control, are demonstrated. Computing loops must be very short in order to reduce current ripple to an acceptable value. A large computing effort is required to achieve a suitable velocity. This requirement is hybrid-control major drawback. It can be overcome by the use of a dedicated FPGA board and a codesign approach to solve partitioning between hardware and software. This approach is used successfully for the hybrid-control algorithm, as reported here, using an FPGA with an embedded processor. Due to parallelism and partitioning of software and hardware, very high execution speed is obtained (less than 10-μs loop). The experimental results validate the usefulness and advantages of codesign solutions for advanced drive control purposes.

When software and hardware partitioning is involved, a codesign methodology is required to obtain an adequate tradeoff that fulfill the awaited specifications of the system in terms of dynamics and the cost constraints that would lead to expensive and over-sized DSPs, for example. Many codesign techniques are reported, and an adaptation to the problem at hand is necessary for design efficiency. FPGAs represent a useful companion to DSPs in industrial context if not an alternative. FPGA embedded processors reveal sufficiently for the software part of most control algorithms, as demonstrated here with an advanced example. The use of microcontroller or DSP circuits is related to the availability of A/D and D/A converters. These functions may be added on an FPGA board, but the overall architecture could fail to meet the cost of commercial processor-based solutions for this sole issue. Moreover, isolation capabilities are generally required between the control board and the power converter. A companion interface board is then necessary, and the question of A/D and D/A converters is a little bit eluded compared to the efficiency offered by mixed hardware/software implementation of control algorithm.

References

[1] I. Takahashi and T. Noguchi, "A new quick response and high efficiency control strategy of an induction motor," *IEEE Trans. Ind. Appl.*, vol. IA-22, no. 5, pp. 820–827, Sep./Oct. 1986.

[2] M. Depenbrock, "Direct self control of inverter-fed induction machines," *IEEE Trans. Power Electron.*, vol. 3, no. 4, pp. 420–429, Oct. 1988.

[3] M. F. Rahman, L. Zhong, M. E. Haque, and M. A. Rahman, "A direct torque-controlled interior permanent-magnet synchronous motor drive without a speed sensor," *IEEE Trans. Energy Convers.*, vol. 18, no. 1, pp. 17–22, Mar. 2003.

[4] G. S. Buja and M. P. Kazmierkowski, "Direct torque control of PWM inverter-fed AC motors—A survey," *IEEE Trans. Ind. Electron.*, vol. 51, no. 4, pp. 744–757, Aug. 2004.

[5] P. Vas, *Sensorless Vector and Direct Torque Control*. London, U.K.: Oxford Univ. Press, 1998. 560 p.

[6] M. S. Branicky, V. S. Borkar, and S. K. Mitter, "A unified framework for hybrid control: Model and optimal control theory," *IEEE Trans. Autom. Control*, vol. 43, no. 1, pp. 31–45, Jan. 1998.

[7] Y. Y. Tzou and T. S. Kuo, "Design and implementation of all FPGA based motor control IC for permanent magnet ac servo-motors," in *Proc. IEEE Int. Conf. Ind. Electron., Control and Instrum.*, 1997, vol. 2, pp. 943–947.

[8] B. Grabergs and B. Sohlberg, "Implementation of a state-space controller in a FPGA," in *Proc. Mediterranean Electrotechnical Conf.*, 1998, vol. 1, pp. 566–569.

[9] B. Bomar, "Implementation of a microprogrammed control in FPGAs," *IEEE Trans. Ind. Electron.*, vol. 49, no. 2, pp. 415–422, Apr. 2002.

[10] Y. Y. Tzou, H. J. Hsu, and T. S. Kuo, "FPGA-based SVPWM control IC for 3-phase PWM inverter," in *Proc. IEEE Int. Conf. Ind. Electron., Control and Instrum.*, 1996, vol. 1, pp. 138–143.

[11] S. Bolognani, M. Tomasini, L. Tubiana, and M. Zigliotto, "DSP-based time optimal current control for high dynamic IPM motor drives," in *Proc. IEEE 35th Annu. PESC*, Jun. 20–25, 2004, vol. 3, pp. 2197–2203.

[12] A. Rubaai, A. Ofoli, and D. Cobbinah, "DSP based real-time implementation of an adaptive fuzzy controller for the tracking control of servo-motor drives," in *Conf. Rec. IAS Annu. Meeting*, Oct. 2–6, 2005, vol. 2, pp. 1317–1323.

[13] Z. Bielewiez, L. Debowski, and E. Lowiec, "A DSP and FPGA based integrated controller development solution for high performance electric drives," in *Proc. IEEE Int. Conf. Ind. Electron.*, 1996, vol. 2, pp. 679–684.

[14] W. Wolf, "A decade of hardware/software co-design," *Computer*, vol. 36, no. 4, pp. 38–43, Apr. 2003.

[15] M. Kharrat, N. Masmoudi, M. Ghariani, and L. Kamoun, "A digital control system based on co-design methodology," in *Proc. 10th Int. Conf. Microelectron.*, 1998, pp. 257–261.

[16] P. Dhaussy, J. Filloque, B. Pottier, and S. Rubini, "Global control synthesis for a MIMD/FPGA machine," in *Proc. IEEE Workshop FPGA Custom Comput. Mach.*, 1994, pp. 72–81.

[17] Actel. (2006). *Actel Fusion—The World's First Mixed-Signal FPGA*. [Online]. Available: http://www.actel.com/products/

[18] *DS1104 Installation and Configuration Guide,* Dspace GmbH, Paderborn, Germany, Mar. 2005.

[19] *Quartus II v2.2 Hand Book*, Altera, San Jose, CA, 2004.
[20] Altera. (2004). *Nios Embedded Processor Software Development Reference Manual*. [Online]. Available: http://www.altera.com/literature/manual
[21] *SOPC Builder v3.0 Reference*, Altera, San Jose, CA, 2004.
[22] *ModelSim Users' Guide*, Mentor Graphics, Wilsonville, OR, 2004.
[23] *Leonardo Spectrum Tools*, Mentor Graphics, Wilsonville, OR, 2004.
[24] X. Lin-Shi. (2006). *Motor Control Using a FPGA-Based System Application to So-Called Hybrid Control Algorithm*. [Online]. Available: http://klimt2.insa-lyon.fr/hybrid/

Xuefang Lin-Shi received the Ph.D. degree in applied computer science and automatics from the Institut National des Sciences Appliquees de Lyon (INSA Lyon), Villeurbanne, France, in 1992.

She was with the Electrical Engineering Center, Lyon (CEGELY), in 1994. Since 1994, she has been an Assistant Professor in the Electrical Engineering Department, INSA Lyon. She is currently with the Laboratoire Ampère, Lyon, France. Her research interests concern control applied to electrical drives and, recently, to hybrid dynamic systems.

Florent Morel (S'06) was born in Bourg-en-Bresse, France, in 1978. He graduated from the École Normale Supérieure de Cachan, Cachan, France. He received the Agrégation in electrical engineering in 2002 and the M.S. degree in electrical engineering from the Institut National de Sciences Appliquées, Lyon, France, in 2003. He is currently working toward the Ph.D. degree in the Laboratoire Ampère, INSA Lyon, Lyon.

His research concerns control applied to power electronics systems.

Ana M. Llor received the M.S. degree from the Escuela Politécnica Superior de la Universidad Carlos III, Madrid, Spain, in 1998, and the Ph.D. degree from the Escuela Politécnica Superior de la Universidad Carlos III, Madrid, Spain, and the Institut National des Sciences Appliquées, Lyon, France, in 2003.

She was with the Laboratoire d'Electrotechnique et d'Electronique Industrielle, France, as an Assistant Professor, in 2005. Her research interests include hybrid control of power converters, digital control of electrical systems, and series multicell converters.

Bruno Allard (M'93–SM'02) received the Engineer, M.Sc., and Ph.D. degrees from the Institut National des Sciences Appliquées de Lyon (INSA Lyon), Lyon, France, in 1988, 1989, and 1992, respectively.

He was with the Centre de Génie Electrique de Lyon, Lyon, as an Associate Professor, in 1992, where he became an Assistant Professor, in 1993. He is currently with the Laboratoire Ampère, INSA Lyon, Lyon. His research interests include power semiconductor-device modeling and characterization, power-electronic-system design, monolithic converter design either for silicon low-power applications or silicon-carbide high-temperature applications.

Jean-Marie Rétif received the Ph.D. degree in tuning control from the Claude Bernard University, Lyon, France, in 1979.

He was with the Laboratory of Thermodynamics and Control, Institut National des Sciences Appliquées de Lyon, where he dealt with different subjects such as process control and optimization of solar heating. Since 1979, he has been working on the control of electrical engineering systems at the Laboratoire Ampère, INSA Lyon, Lyon. Currently, he is with the Laboratiore Ampère, INSA Lyon, Lyon, as a Professor. He is interested in the control of hybrid dynamic systems and applications concerning power systems.